IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant(s):	Swedor et al.	Examiner:	Reilly, Sean
Serial No. :	09/692,949	Group Art No.:	2153
Filed :	October 20, 2000		
Atty Docket :	120-078		
Title :	Method and Apparatus for Using Documents Written in a Markup Language to Access and Configure Network Elements		

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF PURSUANT TO 37 C.F.R. § 1.192

This Appellant's brief is hereby submitted in accordance with a Notice of Appeal filed on August 8, 2006.

I. **Real Party in Interest**

The real party in interest is Nortel Networks, Limited.

II. **Related Appeals and Interferences**

Appellants are not aware of any appeals or interferences that are related to the present case.

III. **Status of the Claims**

This is an appeal brief from a decision by the Primary Examiner dated April 7, 2006, finally rejecting all the claims currently pending in the present application. No claims have been allowed. The currently pending claims are 1-43 and 45-50. Claim 44 was cancelled in the Amendment filed January 23, 2006.

The rejections of claims 1-43 and 45-50 are the subject of this appeal.

A notice of Appeal was filed on August 8, 2006.

IV. **Status of Amendments**

No amendments to the claims were filed after the Final Rejection of April 7, 2006. As noted above, Claim 44 was cancelled in the Amendment filed January 23, 2006.

V. **Summary of Claimed Subject Matter**

The subject matter of independent claims 1, 17, 33, 34 and 48 sets forth methods and devices for controlling a data forwarding service in a data forwarding device. The controlling of the data forwarding service in the independent claims 1, 17, 33, 34 and 48 includes receiving a document written in accordance with a markup language and a corresponding document definition, wherein the document describes the data forwarding service by specifying a class of objects for the data forwarding service. This aspect of the claimed invention is disclosed at page 2 line 19 through page 3 line 11 of the Specification, and illustrated in Figure 7 at step 702. The independent claims 1, 17, 33, 34 and 48 further include parsing of the received document in accordance with the corresponding document definition, wherein the parsing determines at least one parameter describing the data forwarding service, as disclosed in the Specification on page 3 lines 1-3, and from page 11 line 13 through page 14 line 2, and also illustrated by parser 402 in Figure 4 and the parsing step 704 in Fig. 7.

The step of executing the data forwarding service on the network device, as set forth in the independent claims 1, 17, 33, 34 and 48, upon completion of the parsing, in accordance with the parsed document, is disclosed in the Specification on page 3, lines 3-5. The executing of the data forwarding service in independent claims 1, 17, 33, 34 and 48 includes instantiating and launching the data forwarding service in the data forwarding device based on the class of objects for the data forwarding service, as illustrated at step 706 of Figure 7, and as described on page 14, lines 3-16 of the Specification. The data forwarding service of independent claims 1, 17, 33 and 34 configures a forwarding architecture in the network device operable to filter network traffic, as described in the Specification on page 6 lines 3-4, page 7, lines 14-15, page 8 line 11, page 9 lines 2-5, and at page 10 line 18 through page 11 line 3.

The subject matter of dependent claims 2 and 18 provides that the executing of the data forwarding service includes interfacing with hardware and software on the network device. Examples of such embodiments are disclosed in the Specification on page 3 lines 5-6, page 7 lines 14-15, page 17 lines 16-17, and at lines 8-9 of page 19.

The subject matter of dependent claims 3, 19, 36 and 49 sets forth that the markup language is XML, as disclosed in Specification at line 16 on page 2, and at line 5 on page 3. That the corresponding document definition is an XML DTD, as in dependent claims 4, 20, 37 and 50, is disclosed in the Specification on page 3 at line 3.

Dependent claims 5, 21 and 38 provide for retrieving the corresponding definition from a plurality of document definitions in accordance with an identifier in the received document, as is disclosed at page 7, lines 1-6 of the Specification.

Dependent claims 6, 22 and 38 set forth that the plurality of document definitions are provided in a local storage of the network device, which is disclosed in the Specification on page 7 line 20, and on page 8 at lines 3-7.

Dependent claims 7, 23 and 39 include retrieving the corresponding document definition from a plurality of document definitions in accordance with an identifier in the received document, as is disclosed by the teachings at page 11 of the Specification, in lines 15-17. The subject matter of dependent claims 8, 24 and 40 sets forth that the plurality of document definitions are provided in a local storage of the network device, as disclosed in the Specification at page 7, line 20, and page 8, lines 3-7.

The subject matter of dependent claims 9, 25 and 41 sets forth that the parsing includes parsing from the document an identifier corresponding to the service, as described in the Specification from page 11 line 5, through page 14 line 7. The subject matter of dependent claims 10, 26 and 42, which provides that the parsing further includes the step of parsing from the document runtime parameters corresponding to the service, is also supported by those sections of the Specification.

Dependent claims 11, 27 and 43, which include instantiating an object corresponding to the service in accordance with the parsed identifier, and dependent claims 12 and 28, which include instantiating an object corresponding to the service in accordance with the parsed identifier and the parsed runtime parameters, are supported by the Specification in lines 4-7 on page 14.

The subject matter of dependent claims 13 and 29, in which the network device is made up of a router, a switch, or a hub, is disclosed on page 6, lines 3-4, and illustrated by device 104' of Figures 2 and 3.

Dependent claims 14 and 30 provide that the network device includes a forwarding architecture, as disclosed at line 18 of page 10 through line 3 of page 11 of the Specification. Dependent claims 15 and 31 set forth preparing a response corresponding to the executed service, and dependent claims 16 and 32 set forth forwarding the response to a remote requestor of the service. These claims are described in the Specification on page 11 at lines 4-6, and page 16 line 9 through page 17 line 6, and are illustrated by steps 710 through 714 in Figure 7.

Dependent claim 35 sets forth that the network data transfer service includes an HTTP server, as in the Specification on page 10, lines 1-2, and shown as 302 in Figure 3.

Dependent claim 39 provides a services storage coupled to the service launcher that stores a plurality of services, the service launcher being further adapted to select the service from the stored plurality of services in accordance with the parsed identifier, as described in the Specification on page 9, lines 8 through 12, and shown in Figures 3 and 4 by Services 308.

Dependent claim 40 further includes an Oplet Runtime Environment, the service launcher being further adapted to launch the service under the Oplet Runtime Environment, as described in the Specification on page 9 at lines 8-20.

Dependent claim 41 further includes a packet forwarding switch fabric, as shown by switch fabric 604 in Figure 6. Dependent claim 42 provides that the launched service causes changes in how packets are forwarded through the packet forwarding switch fabric, as is described on page 9 at lines 2 through 8.

The subject matter of dependent claim 43 involves a service that monitors performance indicators of how packets are forwarded through the packet forwarding switch fabric, as described on page 9, lines 6-8 of the Specification.

Claims 45, 46 and 47 provide APIs for interoperating with device hardware and software for executing the launched services, as illustrated by APIs 406 in Figure 4, and APIs 504 in Figures 5 and 6.

VI. Grounds of Rejection to be Reviewed on Appeal

A. Claims 1-4, 9-10, 12-20, 25-26, 28-37, 39, 41-44, and 46-47 stand rejected as obvious under 35 U.S.C. 103(a) over the combination of "XNAMI - An eXtensible XML-based paradigm for Network and Application Management Instrumentation" by John et al. ("John et al.") with U.S. Patent Number 5,541,911 of Nilakantan et al. ("Nilakantan et al.").

B. Dependent claims 5-8, 11, 21-24, 27, 38 and 48-50 stand rejected for obviousness under 35 U.S.C. 103(a) based on John et al. and Nilakantan et al., in further combination with "An Introduction to the Extensible Markup Language XML" by Bryan ("Bryan").

C. Claims 40 and 46 stand rejected for obviousness under 35 U.S.C. 103(a), based on the combination of John et al. and Nilakantan et al. with lines 8-16 on page 9 of the Applicants' Specification, and "Dynamic Classification in Silicon-based Forwarding Engine Environments" (Jaeger).

VII. Argument

A. The Examiner has failed to establish a *prima facie* case of obviousness under 35 U.S.C. §103(a) in the rejection of claims 1-4, 9-10, 12-20, 25-26, 28-37, 39, 41-44, and 46-47 using the combination of John et al. with Nilakantan et al.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Appellants assert that the combination of John et al. with Nilakantan et al. fails to disclose or suggest the claimed limitation of the present independent claims 1, 17 and 33, which provides ***a description of a data forwarding service by specification of a class of objects for the data forwarding service in a markup language document, and instantiates and launches the data forwarding service on a network device based on the class of objects from the markup language document***. In contrast, John et al. and Nilakantan et al. describe techniques for defining and/or manipulating MIB table variables in order to control services on a network device that operate in response to the MIB. Additionally, if an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Appellants accordingly assert that the dependent claims 2-4, 9-10, 12-16, 18-20, 25-26, 28-32, 34-37, 39, 41-44, and 46-47 are also nonobvious over the combination of John et al. and Nilikantan et al. based on the nonobviousness of claims 1, 17, and 33 over the combination of John et al. and Nilikantan et al.

<u>John et al., "XNAMI - An eXtensible XML-based paradigm for Network and Application Management Instrumentation":</u>

<u>John et al.</u> discloses a system in which XML is used to convey a new MIB variable definition, together with code operable to process subsequent GET and SET requests for that MIB variable. This teaching of <u>John et al.</u> is well illustrated in the SNMP PDU containing the "XML string for variable definition" and "Compressed Method bytecode" shown in Fig. 8 of <u>John et al.</u> on page 7. With regard to the "XML string for variable definition" shown in Fig. 8, <u>John et al.</u> state as follows beginning at line 32 of the left hand column of page 7:

> The value to which the manager SETs mib_proxy is a string containing a description in XML of the new objects in the subtree. (emphasis added)

<u>John et al.</u> go on to describe the "Compressed Method bytecode" in the SNMP PDU of Fig. 8 as follows beginning at line 3 of the right hand column of page 7:

> The values which the manager passes to the methods_proxy object are strings containing the compressed Java byte code, one for each leaf node being added to the tree. The bytecode strings contain the code for doing GET and SET operations on the newly-added MIB objects. As the new MIB objects are created, the bytecode for each node is decompressed and loaded as a Java class containing two methods, one for a GET on the MIB object and one for a SET. (emphasis added)

The above sections of John et al. teach that the Java class defined by the SNMP PDU of Fig. 8 indicates methods for GET and SET operations on a newly created MIB variable. As described in Section 2.2 of John et al., SNMP SET operations are modification operations on MIB variables, while SNMP GET operations are a type of retrieval operations on MIB variables.

Nilakantan et al., U.S. Patent Number 5,541,911:

Nilakantan et al. teaches filtering operation based on MIB variables. Fig. 3 of Nilakantan et al. shows a Smart Filter master and a Smart Filter agent. As described in columns 5 and 6 of Nilakantan et al., Smart Filter master code is implemented within a central router, which includes an interface to an SNMP transport mechanism that is coupled to a port for communication with the Smart Filter agent over a network. The Smart Filter agent of Nilakantan et al. "includes an SNMP management information base MIB." The Smart Filter agent performs packet spoofing and packet filtering for a LAN in response to information in the MIB.

Claims 1-4, 9-10, 12-20, 25-26, 28-37, 39, 41-44, and 46-47:

Each of the present independent claims 1, 17 and 33 include the limitation of ***a description of a data forwarding service by specification of a class of objects for the data forwarding service in a markup language document, and***

instantiates and launches the data forwarding service on a network device based on the class of objects from the markup language document. Appellants respectfully submit that the combination of John et al. and Nilikantan et al. lacks any disclosure or suggestion of such a feature. In contrast, the combination John et al. and Nilakantan et al. results in a system that uses the techniques in John et al. for conveying *a new MIB variable definition, together with code operable to process subsequent GET and SET requests for that MIB variable*, to another device storing a MIB *used by* the Smart Filter of Nilakantan et al. to perform packet spoofing and packet filtering for a LAN. Nothing in the combination of John et al. and Nilakantan et al. provides any hint or suggestion of even the desirability of controlling a data forwarding service in a network device by:

> receiving at the network device a document written in accordance with a markup language and a corresponding document definition, ***wherein the document describes the data forwarding service by specifying a class of objects for the data forwarding service***;
>
> . . .
>
> executing the data forwarding service on the network device . . . wherein the executing includes ***instantiating and launching the data forwarding service in the data forwarding device based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device to filter network traffic***. (emphasis added)

as in the present independent claim 1 (analogous features are present in independent claims 17 and 33). The only mark-up language described in the combination of John et al. and Nilakantan et al. is the XML document describing the MIB (page 121, column 1, lines 1-8), and the XML string for variable definition shown in Fig. 8 of John et al. Neither of these descriptions includes any teaching of a markup language document that *describes a data forwarding service by specifying a class of objects for the data forwarding service, nor any instantiating and launching of the data forwarding service based on the class of objects in the data forwarding service*, as in the present independent claims 1, 17 and 33. As a result, the combination of John et al. and Nilakantan et al. fails to allow dynamic instantiation and launching of a service based on an object class specifying the service contained in a markup language document, which is an advantage of the present independent claims 1, 17 and 33 over such systems. In contrast, John et al. and Nilakantan et al. are limited to providing GET and SET functions and definitions for MIB variables, and then using the resulting MIB to control the Smart Filter Agent of Nilakantan et al.

In the Advisory Action dated July 12, 2006, the Examiner further states as follows:

> … John teaches much more than Applicant contents. In particular John does not merely provide code for performing SET and GET methods on MIB variables. The entire point of John's system is to impart new functionality in network devices being managed by enabling users (such as network managers or administrators) to define and upload MIB variables to managed network devices wherein the MIB variables implement a new function (see Inter alia the discussion on pg 2, Col 1 and 2 and also section 4). For instance an administrator may add a new MIB variable that tracks total CPU and total memory usage at the managed network device (see pg 8, Col 2). The functionality of new MIB variables in John's system is purposely left open ended so that administrators and other uses can add new functionality to devices as they wish. . .

Applicants do not dispute that the teachings of John et al. extend to managing a network element based on the additional MIB variables that can be accessed using the GET and SET processing code shown in the SNMP PDU of Fig. 8 of John et al. But this extended management provided by John et al. is with regard to the retrieval and presentation of the added MIB variables. The need for these functions is set forth in section 1 of John et al., which describes the problem occurring "where the manager requests for the value of variable *X* but gets back a reply stating that *X* is not in the MIB". Later in section 1, John et al. describes the technology disclose therein as "a new paradigm which radically changes the retrieval and presentation of management data - an XML-based *dynamically reconfigurable* or *eXtensible* MIB" (italics in original). Section 4 of John et al. accordingly concerns the addition of new MIB variables, deletion of unused MIB variables, provision of a view of the MIB, and selection of MIB variables for monitoring.

The fact that John et al. provides for a way to change how a network device or element is managed by allowing for defining and uploading new MIB variables does not mean that John et al. teaches the specific approaches to instantiating and launching a data forwarding service that are features of the present independent claims 1, 17 and 33. Applicants are not attempting to obtain claims covering any and all ways of extending the management capabilities of a network device, and the language of the present independent claims 1, 17 and 33 is not so broad as that. Instead, the present independent claims 1, 17 and 33 are specifically directed in significant part to *receiving a document written in accordance with a markup language and a corresponding document definition, wherein the document describes a data forwarding service by specifying a class of* *objects for the data forwarding service, and executing the data forwarding service on the network device, wherein the executing includes instantiating and launching the data forwarding service in the data forwarding device based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device to filter network traffic*. John et al. falls short not with regard to the possible "functionality of the new MIB variables" that it supports, but rather with regard to its failure to disclose or suggest the above highlighted features of the present independent claims 1, 17 and 33. Moreover, neither the fact that the MIB of John et al. is provided as a tree of Java object, nor that new MIB objects can be added to the MIB tree in John et al. remedy this deficiency.

Further in the Advisory Action, the Examiner emphasizes columns 5 and 6 of Nilikantan et al. As noted above, the teaching of Nilikantan et al. in that section only describes the possibility of performing packet filtering and/or packet spoofing "in response to information in the MIB". Accordingly, the addition of Nilikantan et al. to John et al. still provides no hint or suggestion of the above indicated features of independent claims 1, 17 and 33.

For these reasons, the combination of John et al. and Nilakantan et al. does not disclose or suggest all the limitations of the present independent claims 1, 17 and 33. John et al. and Nilakantan et al. accordingly do not form a prima *facie case* of obviousness with regard to the present independent claims 1, 17 and 33 under 35 U.S.C. 103. As to dependent claims 2-4, 9-10, 12-16, 18-20, 25-26, 28-32, 34-37, 39, 41, and 44-47, they each depend from claims 1, 17 and 33. As such, Applicants respectfully assert that these dependent claims are also patentable over the combination of John et al. and Nilakantan et al. for at least the same reasons.

B. The Examiner has failed to establish a *prima facie* case of obviousness under 35 U.S.C. §103(a) in the rejection of claims 5-8, 11, 21-24, 27, 38 and 48-50 using the combination of John et al., Nilikantan et al., and Bryan.

As noted above, in order to establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. Appellants have argued above that the combination of John et al. with Nilakantan et al. does not disclose or suggest the claimed limitation of the present independent claims 1, 17 and 33, which provides ***a description of a data forwarding service by specification of a class of objects for the data forwarding service in a markup language document, and instantiates and launches the data forwarding service on a network device based on the class of objects from the markup language document***. This feature is also present in independent claim 48. Applicants now respectfully urge that the addition of Bryan to John et al. and Nilikantan et al. does not remedy this deficiency. Accordingly, the dependent claims 5-8, 11, 21-24, 27, 38 and 49-50 are also nonobvious over the combination of John et al., Nilikantan et al. and Bryan based on the nonobviousness of independent claims 1, 17, 33 and 48 over the combination of John et al., Nilikantan et al., and Bryan.

Bryan, "An Introduction to the Extensible Markup Language XML":

Bryan provides an overview of commonly used components in XML, including Document Type Definitions (DTDs) to formally identify the relationships between elements in an XML document.

Claims 5-8, 11, 21-24, 27, 38 and 48-50:

As with John et al. and Nilakantan et al., nowhere in Bryan is there disclosed or suggested any method or system for controlling a data forwarding service in a network device comprising a data forwarding device, as in the present independent claims 1, 17, 33 and 48, from which claims 5-8, 11, 21-24, 27, 38 and 49-50 depend. Neither John et al., Nilakantan et al., nor Bryan teach or suggest:

> receiving at the network device a document written in accordance with a markup language and a corresponding document definition, ***wherein the document describes the data forwarding service by specifying a class of objects for the data forwarding service***;
>
> . . .
>
> executing the data forwarding service on the network device . . . wherein the executing includes ***instantiating and launching the data forwarding service in the data forwarding device based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device to filter network traffic***. (emphasis added)

Neither John et al., Nilakantan et al. nor Bryan include even a suggestion of the desirability of the above indicated features of the present independent claims, from which claims 5-8, 11, 21-24, 27, 38 and 49-50 depend. The shortcomings of John et al. and Nilikantan et al. in this regard are discussed above. While Bryan does provide teachings regarding the general use of DTDs in XML, those teachings do not address the above cited features of the present independent claims.

Applicants therefore respectfully urge that the combination of John et al., Nilakantan et al., and Bryan does not disclose or suggest all the limitations of the present independent claims 1, 17, 33 and 48. The combination of John et al., Nilakantan et al. and Bryan accordingly does not support a *prima facie* case of obviousness under 35 U.S.C. 103 with regard to the present independent claims 1, 17, 33 and 48. As to dependent claims 5-8, 11, 21-24, 27, 38 and 49-50, they each depend from claims 1, 17, 33 and 48, and are respectfully believed to be patentable over the combination of John et al., Nilakantan et al. and Bryan for at least the same reasons.

C. The Examiner has failed to establish a *prima facie* case of obviousness under 35 U.S.C. §103(a) in the rejection of dependent claims 40 and 46 using the combination of John et al. and Nilakantan et al., as well as lines 8-16 on page 9 of the Applicants' Specification, and "Dynamic Classification in Silicon-based Forwarding Engine Environments" (Jaeger).

As discussed above, the combination of John et al. with Nilakantan et al. does not disclose or suggest the claimed limitations of the present independent claim 33, from which 40 and 46 depend, which include ***a description of a data forwarding service by specification of a class of objects for the data forwarding service in a markup language document, and instantiates and launches the data forwarding service on a network device based on the class of objects from the markup language document***. Applicants respectfully urge that the addition of Jaeger to John et al. and Nilikantan et al. does not remedy this deficiency of John et al. and Nilikantan et al. in this regard, and that the cited sections of the Applicants' Specification are not prior art.

Lines 8-16 on page 9 of Applicants' Specification:

Applicants assert that the statements in lines 8-16 on page 9 of the Specification do not constitute an admission of prior art, at least in part because the first words of that paragraph are "In one example of the invention. . . " Additionally, the section of the Specification containing those lines is entitled "DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS". Moreover, the cited text repeatedly refers to the ORE^{TM} embodiment as "an example of the invention".

While the use of ORE (Oplet Runtime Environment) may represent one aspect of a possible embodiment of the presently claimed invention, its description as such in the Applicants' own disclosure is clearly intended as part of what the Applicants believe to be their invention. The use of portions of the Applicants' own Detailed Description unfairly and inappropriately uses the Applicants' own teachings to make up for what cannot fairly be found and/or identified in the actual prior art.

For at least these reasons, the statements in lines 8-16 on page 9 of the Specification should not be considered an admission of prior art.

Jaeger, "Dynamic Classification in Silicon-based Forwarding Engine Environments":

Jaeger discloses a programmable network architecture built on a Gigabit Ethernet L3 Routing switch to support downloadable services.

Claims 40 and 46:

As with John et al. and Nilakantan et al., nowhere in Jaeger is there disclosed or suggested any method or system for controlling a data forwarding service in a network device comprising a data forwarding device as in the present independent claim 33, from which claims 40 and 46 depend. Neither John et al., Nilakantan et al., nor Jaeger teach or suggest a network device for locally performing a data forwarding service in accordance with a received document written in a document markup language, wherein the network device comprises a data forwarding device, including:

> a parser that is adapted to ***parse the received document in accordance with a document definition to obtain an identifier of the service, wherein the parsing determines at least one parameter describing the data forwarding service by specifying a class of objects for the data forwarding service***; and
>
> a service launcher that is adapted to ***launch the data forwarding service corresponding to the identifier parsed from the received document, wherein the service launcher instantiates and launches the data forwarding service in the data forwarding device upon completion of the parsing based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device operable to filter network traffic***. (emphasis added)

Jaeger discloses a programmable network architecture built on a Gigabit Ethernet L3 Routing switch to support downloadable services. Neither John et al., Nilakantan et al. nor Jaeger include even a suggestion of the desirability of the above indicated features of the present independent claim 33, from which claims 40 and 46 depend. For these reasons, Applicants respectfully urge that the combination of John et al. and Nilakantan et al. with Jaeger does not disclose or suggest all the limitations of the present independent claim 33, from which dependent claims 40 and 46 depend.

Applicants further submit that the Examiner has not established a sufficient motivation to combine John et al. and Nilakantan et al. with Jaeger. A *prima facie* case of obviousness under 35 U.S.C. 103 must include a showing of a suggestion, teaching or motivation that would have led a person of ordinary skill in the art to combine the cited references in the particular manner claimed. See In re Dembiczak, 175 F.3d 994, 998 (Fed. Cir. 1999), and In re Kotzab, 217 F.3d 1365, 1371 (Fed. Cir. 2000). "[C]ombining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability—the essence of hindsight." Dembiczak, 175 F.3d at 999. In the final rejection, the Examiner again asserted that a skilled person would be motivated to combine John et al. with Jaeger based on teachings in the Conclusion of Jaeger -- that Conclusion states that ORE "supports the creation of services in Java that are extensible, portable, and easily distributed over the network". Applicants respectfully disagree, in light of the fact that the solution described in John et al. concerns providing *definitions of and access to new MIB variables*, and does not point to any need for "creation of services in Java". Accordingly, Applicants respectfully urge that one skilled in the art would not be motivated to modify John et al. to include the teachings of Jaeger et al. for the reasons cited by the Examiner, since the systems of John et al. and Jaeger have different objectives in this regard. Moreover, the Examiner has provided no motivation whatsoever for combining Nilakantan et al. with Jaeger.

The combination of John et al. and Nilakantan et al. with Jaeger accordingly does not support a *prima facie* case of obviousness under 35 U.S.C. 103 with regard to the present independent claim 33, and dependent claims 40 and 46 are respectfully believed to be patentable over the combination of John et al. and Nilakantan et al. with Jaeger for at least the same reasons.

VIII. Conclusion

Appellants submit therefore that the rejections of the present claims under 35 U.S.C. and 103, based on John et al., Nilikantan et al., Bryan, Jaeger, and pages 8-16 of the Applicants' Specification, are improper for at least the reasons set forth above. Appellants accordingly request that the rejections be withdrawn and the pending claims be allowed.

Respectfully submitted,

NORTEL NETWORKS LTD.

By: /David Dagg/
David A. Dagg
Reg. No. 37,809
Attorney for Assignee

Date: October 5, 2006

McGuinness & Manaras LLP
125 Nagog Park
Acton MA 01720
(617) 630-1131

Appendix A - Claims

1. (previously presented) A method for controlling a data forwarding service in a network device comprising a data forwarding device, comprising the steps of:

receiving at the network device a document written in accordance with a markup language and a corresponding document definition, wherein the document describes the data forwarding service by specifying a class of objects for the data forwarding service;

parsing by the network device the received document in accordance with the corresponding document definition, wherein the parsing determines at least one parameter describing the data forwarding service; and

executing the data forwarding service on the network device upon completion of the parsing, in accordance with the parsed document, wherein the executing includes instantiating and launching the data forwarding service in the data forwarding device based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device operable to filter network traffic.

2. (original) A method according to claim 1, wherein the step of executing includes the step of interfacing with hardware and software on the network device.

3. (original) A method according to claim 1, wherein the markup language is XML.

4. (original) A method according to claim 3, wherein the corresponding document definition is an XML DTD.

5. (original) A method according to claim 1, further comprising:
retrieving the corresponding definition from a plurality of document definitions in accordance with an identifier in the received document.

6. (original) A method according to claim 5, wherein the plurality of document definitions are provided in a local storage of the network device.

7. (original) A method according to claim 3, further comprising the step of:
retrieving the corresponding document definition from a plurality of document definitions in accordance with an identifier in the received document.

8. (original) A method according to claim 5, wherein the plurality of document definitions are provided in a local storage of the network device.

9. (original) A method according to claim 1, wherein the step of parsing includes the step of parsing from the document an identifier corresponding to the service.

10. (original) A method according to claim 9, wherein the step of parsing further includes the step of parsing from the document runtime parameters corresponding to the service.

11. (original) A method according to claim 5, further including the step of:
instantiating an object corresponding to the service in accordance with the parsed identifier.

12. (original) A method according to claim 10, further including the step of:
instantiating an object corresponding to the service in accordance with the parsed identifier and the parsed runtime parameters.

13. (original) A method according to claim 1, wherein the network device comprises one of a router, a switch, and a hub.

14. (original) A method according to claim 1, wherein the network device comprises a packet forwarding architecture.

15. (original) A method according to claim 1, further comprising the step of preparing a response corresponding to the executed service.

16. (original) A method according to claim 14, further comprising the step of forwarding the response to a remote requestor of the service.

17. (previously presented) A network device for locally performing a data forwarding service in response to a remote request, wherein the network device comprises a data forwarding device, comprising:

means for receiving at the network device a document written in accordance with a markup language and a corresponding document definition, wherein the document describes the data forwarding service by specifying a class of objects for the data forwarding service;

means for parsing by the network device the received document in accordance with the corresponding document definitions, wherein the parsing determines at least one parameter describing the data forwarding service; and

means for executing the data forwarding service on the network device upon completion of the parsing, in accordance with the parsed document, wherein the means for executing includes means for instantiating and launching the data forwarding service in the data forwarding device based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device operable to filter network traffic.

18. (original) A network device according to claim 17, wherein the means for executing includes means for interfacing with hardware and software on the network device.

19. (original) A network device according to claim 17, wherein the markup language is XML.

20. (original) A network device according to claim 19, wherein the corresponding document definition is an XML DTD.

21. (original) A network device according to claim 17, further comprising:

means for retrieving the corresponding document definition from a plurality of document definitions in accordance with an identifier in the received document.

22. (original) A network device according to claim 21, wherein the plurality of document definitions are provided in a local storage of the network device.

23. (original) A network device according to claim 19, further comprising:
means for retrieving the corresponding document definition from a plurality of document definitions in accordance with an identifier in the received document.

24. (original) A network device according to claim 21, wherein the plurality of document definitions are provided in a local storage of the network device.

25. (original) A network device according to claim 17, wherein the means for parsing includes means for parsing from the document an identifier corresponding to the service.

26. (original) A network device according to claim 25, wherein the means for parsing further includes means for parsing from the document runtime parameters corresponding to the service.

27. (original) A network device according to claim 21, further including:
means for instantiating an object corresponding to the service in accordance with the parsed identifier.

28. (original) A network device according to claim 26, further including:
means for instantiating an object corresponding to the service in accordance with the parsed identifier and the parsed runtime parameters.

29. (original) A network device according to claim 17, wherein the network device comprises one of a router, a switch, and a hub.

30. (original) A network device according to claim 17, wherein the network device comprises a packet forwarding architecture.

31. (original) A network device according to claim 17, further comprising means for preparing a response corresponding to the executed service.

32. (original) A network device according to claim 30, further comprising means for forwarding the response to a remote requestor of the service.

33. (previously presented) A network device for locally performing a data forwarding service in accordance with a received document written in a document markup language, wherein the network device comprises a data forwarding device, comprising:

a parser that is adapted to parse the received document in accordance with a document definition to obtain an identifier of the service, wherein the parsing determines at least one parameter describing the data forwarding service by specifying a class of objects for the data forwarding service; and

a service launcher that is adapted to launch the data forwarding service corresponding to the identifier parsed from the received document, wherein the service launcher instantiates and launches the data forwarding service in the data forwarding device upon completion of the parsing based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device operable to filter network traffic.

34. (original) A network device according to claim 33, further comprising:
a network data transfer service that is adapted to communicate with remote devices for receiving the document.

35. (original) A network device according to claim 34, wherein the network data transfer service comprises an HTTP server.

36. (original) A network device according to claim 33, wherein the markup language is XML.

37. (original) A network device according to claim 36, wherein the document definition is an XML DTD.

38. (original) A network device according to claim 33, further comprising a document definition storage coupled to the parser that stores a plurality of document definitions, the parser being further adapted to select the document definition from the stored plurality of document definitions in accordance with a document definition identifier.

39. (original) A network device according to claim 33, further comprising a services storage coupled to the service launcher that stores a plurality of services, the service launcher being further adapted to select the service from the stored plurality of services in accordance with the parsed identifier.

40. (original) A network device according to claim 33, further comprising an Oplet Runtime Environment, the service launcher being further adapted to launch the service under the Oplet Runtime Environment.

41. (original) A network device according to claim 33, further comprising a packet forwarding switch fabric.

42. (original) A network device according to claim 41, wherein the launched service causes changes in how packets are forwarded through the packet forwarding switch fabric.

43. (original) A network device according to claim 41, wherein the launched service monitors performance indicators of how packets are forwarded through the packet forwarding switch fabric.

44. (canceled)

45. (original) A network device according to claim 33, further comprising device APIs for interoperating with device hardware and software for executing the launched services.

46. (original) A network device according to claim 40, further comprising device APIs for interoperating with device hardware and software for executing the launched services.

47. (original) A network device according to claim 41, further comprising device APIs for interoperating with device hardware and software for executing the launched services.

48. (previously presented) A method for causing a network device to locally perform a data forwarding service, wherein the network device comprises a data forwarding device, comprising the steps of:

identifying the data forwarding service to be performed at a remote client computer;

preparing at the remote client computer a document written in a markup language in accordance with a document definition, the document including an identifier of the service, wherein the document describes the data forwarding service by specifying a class of objects for the data forwarding service;

transmitting the document to the network device;

identifying at the network device the document definition corresponding to the transmitted document;

parsing by the network device the transmitted document in accordance with the corresponding document definition, wherein the parsing determines the class of objects for the data forwarding service and at least one parameter describing the data forwarding service; and

executing the data forwarding related service on the network device upon completion of the parsing, in accordance with the parsed document, wherein the executing includes instantiating and launching the data forwarding service on the data forwarding device based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device operable to filter network traffic.

49. (original) A method according to claim 48, wherein the markup language is XML.

50. (original) A method according to claim 49, wherein the corresponding document definition is an XML DTD.

Appendix B - Evidence Submitted

None.

Appendix C - Related Proceedings

None.